

nRF2401 无线模块

产品说明书

V1.0 – 2009-9-10



本资料由北京百纳信达科技有限公司编写、版权所有

商标咨询

ATMEL 与 AVR 分别是 ATMEL CORPORATION 的注册商标和商标

百纳信达、mcutop.com、bynatec.com 分别是北京百纳信达科技有限公司的商标与域名

为防止损坏您的 AVR 相关工具，避免您或他人受伤，在使用本开发套装前请仔细阅读下面的安全需知，并妥善保管以便所有本产品设备的使用者都可随时参阅。

请遵守本节中所列举的用以下符号所标注的各项预防措施，否则可能对产品造成损害。



该标记表示警告，提醒您应该在使用本产品前阅读这些信息，以防止可能发生的损害。

警告



请勿在易燃气体环境中使用电子设备，以避免发生爆炸或火灾。



请勿在潮湿的环境中使用电子设备，以避免设备损坏。



发生故障时立即拔下所有线缆。

当您发现产品冒烟或发生异味时，请立刻拔下所有与其连接的线缆，切断电源，以避免燃烧。若在这种情况下还继续使用，可能会导致产品的进一步损坏，并使您受伤。

请与我们联系后，将产品寄回给我们维修。



请勿自行拆卸本产品

触动产品内部的零件可能会导致受伤。

遇到故障时，请及时联系我们。

自行拆卸可能会导致其他意外事故发生。



使用合适的电缆线

若要将线缆连接到本设备的插座上，请使用本产品提供的线缆，以保证产品的规格的兼容性。



请勿在儿童伸手可及之处保管本产品

请特别注意防止婴幼儿玩耍或将产品的小部件放入口中。

注意

北京百纳信达科技有限公司可随时更改手册内所记载之硬件与软件规格的权利，而无需事先通知。

北京百纳信达科技有限公司对因使用本产品而引起的损害不承担任何责任。

北京百纳信达科技有限公司已竭尽全力来确保手册内载之信息的准确性和完善性。如果您发现任何错误或遗漏，请与我们联系（见联系方法），对此，我们深表感谢。

1. 概述	1
1.1 基本参数	1
1.2 功能简介	1
2. 硬件模块说明	2
2.1 nRF2401 及其外围电路	2
2.2 电源及指示电路	3
2.3 接口电路	3
3. 软件说明	4
3.1 配置文件	4
3.2 驱动文件	6
4. 常见问题	8

1. 概述

nRF2401 模块以 nRF2401 芯片为核心,提供 1 路的无线发射和 2 路的无线接收功能,该模块留出 10Pin 排针,可以使用 10Pin 排线方便与 Mega16 直插等开发板连接。

1.1 基本参数

板子尺寸: 25mm*57mm (使用板子自身天线)

25mm*35 mm (使用外部天线)

工作电压: 3V 5V 均可

1.2 功能简介

可以进行无线数据发送和接收;

可以开发无线的电子产品: 如无线遥控器、无线麦克、无线耳机等应用。

2. 硬件模块说明

布局如图所示：

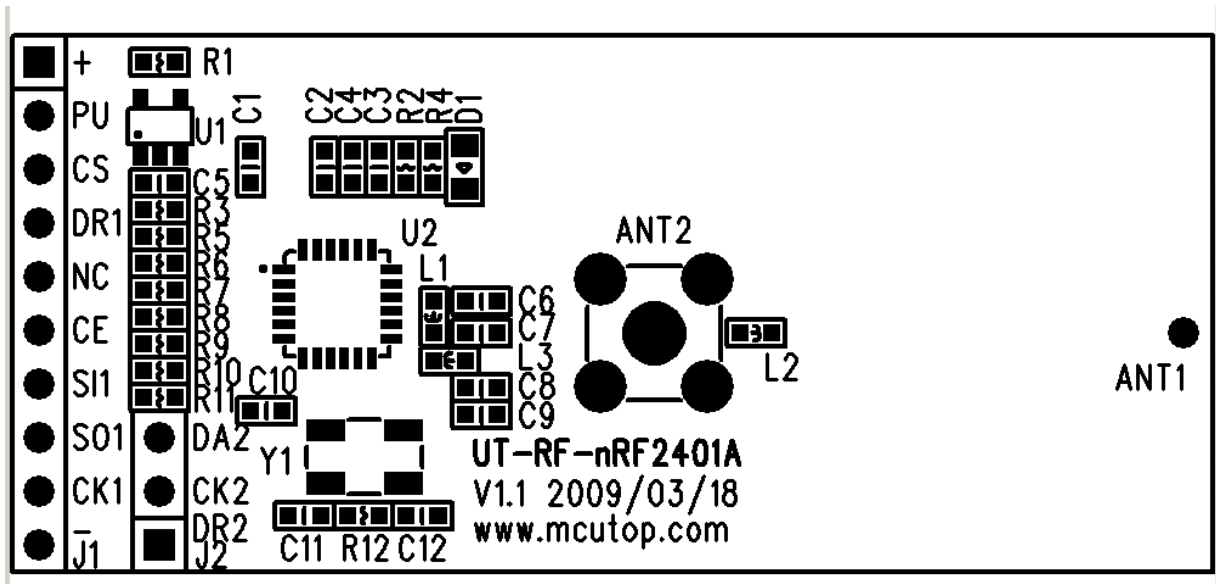


图 1 nRF2401 丝印图

2.1 nRF2401 及其外围电路

nRF2401 及其外围电路如图 2 所示，包括 nRF2401 芯片部分、稳压部分、晶振部分、天线部分。电压 VDD 为芯片提供工作电压；晶振部分包括 Y1、C11、C12，晶振 Y1 允许值为：4MHz、8MHz、12MHz、16MHz，如果需要 1Mbps 的通信速率，则必须选择 16MHz 晶振。天线部分包括电感 L1、L2、L3（板子自身天线）或者 L1、L3、ANT2（外接天线），用来将 nRF2401 芯片 ANT1、ANT2 管脚产生的 2.4G 电平信号转为电磁波信号，或者将电磁波信号转换为电平信号输入芯片的 ANT1、ANT2 管脚。

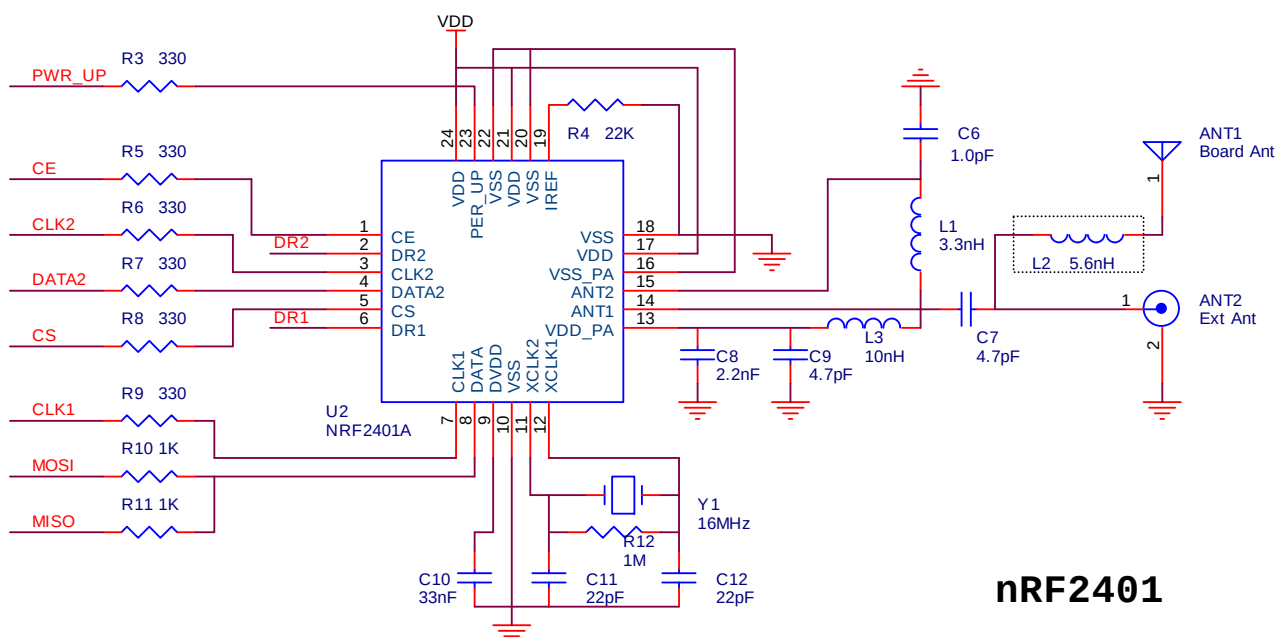


图 2 nRF2401 及其外围电路

2.2 电源及指示电路

电路板上提供 5V 稳压到 3.3V 电路，可以外接 5V。也可以直接接 3.3V，不通过稳压芯片。另外，电路板上提供了一个电源指示灯，当对 nRF2401 无线模块正确供电时，指示灯亮。

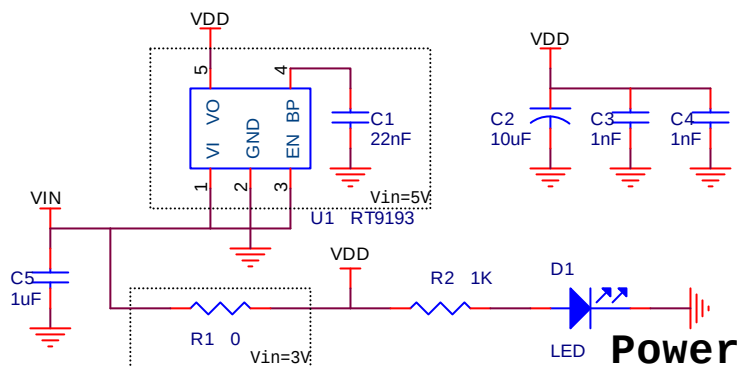


图 3 电源及指示电路

2.3 接口电路

模块提供了两个接口 J1、J2，其中 J1 为 nRF2401A 的控制端口和通道 1 的收发通道，J2 为预留，是通道 2 的接收通道。J1 接口为 10Pin 的排针，非常方便与 Mega16 开发板连接。J2 预留，如果需要采用双通道接收时，可以将 J2 对应的 3 根信号线引出，接到单片机的 I/O 上即可。

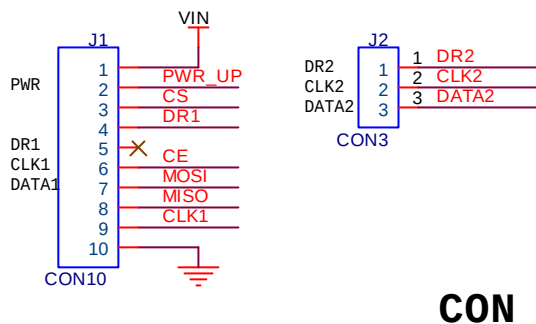


图 4 ISP 接口原理图

3. 软件说明

本模块提供了系统的驱动程序文件 nRF2401.c 和配置文件 nRF2401.h, 包括模块的初始化、配置、数据发送、数据接收等函数。

3.1 配置文件

硬件端口配置, 默认使用 PB 口, 如果更改其他端口修改#define 即可。比如需要使用 PA 口, 那么要做如下修改

```
#define P_NRF2401_PORT    PORTA
#define P_NRF2401_DDR     DDRA
#define P_NRF2401_PIN     PINA
```

对默认定义的控制口 通道 1 使用不同的位, 需要修改#define PWR 即可。例如要使用 Mega16 的 PA2 控制 nRF2401 模块的 PWR 口, 需要做如下修改

```
#define PWR                2           // bit2
```

```
// --- 配置硬件端口 --- //

// 使用mega16 PB口, 使用GPIO可以更改为PA PC PD口
// 使用spi接口, 只能使用PB口
#define P_NRF2401_PORT    PORTB
#define P_NRF2401_DDR     DDRB
#define P_NRF2401_PIN     PINB

#define PWR    0           // bit0 PBO
#define CE     4           // bit4
#define CS     1           // bit1
#define DR1    2           // bit2
#define CLK1   7           // bit7
#define DATA  5           // bit5

#define MOSI   5           // bit5 spi时使用
#define MISO    6           // bit6
```

图 5 硬件端口配置

软件访问 nRF2401 配置:

可以查询 nRF2401 接收到数据, 也可以使用中断方式。当 nRF2401 接收到数据后 DR1 会发出高电平信号, 可以查询该信号也可以使用中断。如果使用中断, 需要将 DR1 信号接到单片机的外部中断引脚。

对 nRF2401 写控制值可以使用 spi 方式, 也可以使用 GPIO 模拟 spi 模式。如果使用 SPI 模式, 需要将 nRF2401 硬件接口与单片机 spi 接口对应连接。

```
// nRF2401的数据接收信号DR1 采用中断还是查询方式检测
// DR1接到mega16的INT2 (PB2) 上
#define NRF2401_POLL 1 // 查询方式
// #define NRF2401_INT 1 // 中断方式

// nRF2401采用gpio模拟spi 或者使用spi接口
// #define NRF2401_GPIO 1 // 采用GPIO模拟
#define NRF2401_SPI 1 // 采用Mega16的spi
```

图 6 硬件端口配置

nRF2401 的配置字

NRF2401_CH1_WIDTH: 通道 1 接收数据宽度, 例如需要一次传 4 个字节, 那么设置为 32

NRF2401_CH1_ADDR1_1: 通道 1 的地址, 接收地址和发送地址需要设置一样

```
// --- nRF2401初始化配置 --- //
#define NRF2401_CH2_WIDTH 0x00 // 通道二接收数据宽度
#define NRF2401_CH1_WIDTH 0x8 // 通道一接收数据宽度

#define NRF2401_CH2_ADDR2_1 0x0 // 通道二地址1
#define NRF2401_CH2_ADDR2_2 0x0
#define NRF2401_CH2_ADDR2_3 0x0
#define NRF2401_CH2_ADDR2_4 0x0
#define NRF2401_CH2_ADDR2_5 0x0

#define NRF2401_CH1_ADDR1_1 0x0 // 通道一地址1
#define NRF2401_CH1_ADDR1_2 0x0
#define NRF2401_CH1_ADDR1_3 0x0
#define NRF2401_CH1_ADDR1_4 0x1
#define NRF2401_CH1_ADDR1_5 0xff
```

图 7 nRF2401 配置字 1

对接收地址宽度、CRC 校验使能、通道 1 还是通道 2 选择、选择直接模式传输还是 BURST 模式传输及速度选择。当选 1M 速度传输, 误码率很加大。

```
#define NRF2401_REC_ADDR_WIDTH (0x20<<2) // 高6个bit 接收地址宽度
#define NRF2401_CRC_LENGTH_8 0x0 // CRC长度
#define NRF2401_CRC_LENGTH_16 0x2
#define NRF2401_CRC_ENABLE 0x1 // CRC使能
#define NRF2401_CRC_DISABLE 0x0

#define NRF2401_RX_CH1 0x00 // 通道1接收
#define NRF2401_RX_CH2 0x80

#define NRF2401_DIRECT_MODE 0x00 // 通信模式 直接
#define NRF2401_SHOCKBURST_MODE 0x40

#define NRF2401_RF_RATE_250 0x00 // 速度
#define NRF2401_RF_RATE_1M 0x20
```

图 8 nRF2401 配置字 2

选择晶振，

—— 全面 快速 免费的数据表查询服务 ——

1

分贝选择，选择-20dB 功率小，传输距离短；选择 0dB 功率大，传输距离远，但是耗电。

NRF2401_FRE_CH_SEL 是对传输频率进行选择，nRF2401 可以在 2.4G~2.5G 之中进行选择。可以有 128 中频率选择

最后的配置字是选择 nRF2401 模块工作在发送模式还是接收模式

```
#define NRF2401_CRYSTAL_4M      (0x0<<2)    // 4MHz
#define NRF2401_CRYSTAL_8M      (0x1<<2)
#define NRF2401_CRYSTAL_12M     (0x2<<2)
#define NRF2401_CRYSTAL_16M     (0x3<<2)
#define NRF2401_CRYSTAL_20M     (0x4<<2)

#define NRF2401_RF_POWER_f20dB   0x0        // -20dB
#define NRF2401_RF_POWER_f10dB   0x1
#define NRF2401_RF_POWER_f5dB    0x2
#define NRF2401_RF_POWER_0dB     0x3

#define NRF2401_FRE_CH_SEL        (0x32<<1)    // 2.45GHz

#define NRF2401_TRANSMIT_MODE 0x0
#define NRF2401_RECEIVE_MDOE  0x1
```

图 9 nRF2401 配置字 3

3.2 驱动文件

需要配置不同工作模式 需要修改 InitData 这个数组

```
const uint8_t InitData[15] =                // nRF2401 控制字 15 Byte
{
    NRF2401_CH2_WIDTH,                       // 通道二接收数据宽度
    NRF2401_CH1_WIDTH,                       // 通道一接收数据宽度

    NRF2401_CH2_ADDR2_1,                     // 通道二地址
    NRF2401_CH2_ADDR2_2,                     // 通道二地址
    NRF2401_CH2_ADDR2_3,                     // 通道二地址
    NRF2401_CH2_ADDR2_4,                     // 通道二地址
    NRF2401_CH2_ADDR2_5,                     // 通道二地址

    NRF2401_CH1_ADDR1_1,                     // 通道一地址
    NRF2401_CH1_ADDR1_2,                     // 通道一地址
    NRF2401_CH1_ADDR1_3,                     // 通道一地址
    NRF2401_CH1_ADDR1_4,                     // 通道一地址
    NRF2401_CH1_ADDR1_5,                     // 通道一地址
```

(NRF2401_REC_ADDR_WIDTH

—— 全面 快速 免费的数据表查询服务 ——

| NRF2401_CRC_ENABLE), // b23~b18:接收地址宽度 (32 位)
// b17~b16: CRC 模式选择(11-16bit)

(NRF2401_RX_CH1
| NRF2401_SHOCKBURST_MODE
| NRF2401_RF_RATE_250
| NRF2401_CRYSTAL_16M
| NRF2401_RF_POWER_0dB), // b15: 接收通道选择, 通道 1 接收
// b14: 收发模式(1-ShockBurst 模式)
// b13: 波特率选择(0-250Kbps)
// b12~b10: 晶振选择(011-16M)

(NRF2401_FRE_CH_SEL
| NRF2401_TRANSMIT_MODE) // b7~b1: 收发频率 (0110010, 2.45GHz)
// b0: 收发控制(1-接收)

};

用户的 API 函数:

```
// --- nRF2401 用户 API 函数 --- //
void nRF2401_ON(void); // nRF2401A 上电
void nRF2401_OFF(void); // nRF2401A 掉电
void nRF2401_Initial(void); // nRF2401A 初始化
void nRF2401_WriteCommand(void); // nRF2401A 写命令字
void nRF2401_WriteByte(uint8_t uiData); // 向 nRF2401A 写入一个 Byte

void nRF2401_Mode(uint8_t uiMode); // 切换 nRF2401A 收/发模式

void nRF2401_SendByte(uint8_t TxData); // 发送 1 字节数据
void nRF2401_SendWord(uint16_t TxData); // 发送 1 字数据

// 将数组数据写入 nRF2401A
void nRF2401_SendBuffer_Byte(uint8_t TxData[], uint8_t DataByte);

// 将数组数据写入 nRF2401A
void nRF2401_SendBuffer_Word(uint16_t TxData[], uint8_t DataWord);
```

uint8_t nR

—— 全面 快速 免费的数据表查询服务 ——

查询方式

```
void nRF2401_ReceiveByte(uint8_t RxData[]);    // 以 Byte 格式从 nRF2401A 读数据
```

```
void nRF2401_ReceiveWord(uint16_t RxData[]);    // 以 Word 格式从 nRF2401A 读数据
```

```
// 设置目标 nRF2401A 的地址
```

```
void nRF2401_SetAddress(uint8_t Address[], uint8_t AddrByte);
```

4. 常见问题

为什么通讯不成功？

重点检查

发送端发送的地址和接收端接收的地址是否一致（包括地址值和有效位）

发送端发送的数据宽度和接收端接收的设置是否一致

发送端的发送频率和接收端接收频率是否一致

联系方式

网站: <http://www.mcutop.com>论坛: <http://bbs.mcutop.com>邮箱: support@mcutop.com

qq 客服: 1059098245